

CS 6410: Compilers

Fall 2019

HW 2 – Context Free Grammars and LR Parsing

Assigned: Tuesday, October 1, 2018, **Due: Saturday, October 19, 2019**

Instructor: Tamara Bonaci
College of Computer and Information Science
Northeastern University – Seattle

Submission Guidelines

- Please push your homework as a single .pdf file through Canvas.
- You do not have to type in your submission - hand-written and then scanned, or photographed documents are fine, as long as the total size of your document is not too big, and your document is readable.
- This assignment is meant to be worked on individually, and you should submit it by **11:59pm on Saturday, October 19, 2019.**

Problem 1 (Problem modified from the *Aho, Sethi, Ullman book*)

Please consider the following grammar:

$S ::= cSdS \mid dScS \mid \epsilon$

1. Show that this grammar is ambiguous by constructing **two leftmost derivations** for the sentence $cdcd$.
2. Show that this grammar is ambiguous by constructing **two rightmost derivations** for the sentence $cdcd$.
3. Construct the corresponding parse trees for your derivations.

Problem 2 (Problem modified from the *Aho, Sethi, Ullman book*)

Consider the following grammar:

$S ::= (L) \mid x$

$L ::= L , S \mid S$

1. Give a left-most derivation of $(x, (x, x))$.
2. Give a right-most derivation of $(x, (x, x))$.
3. Show the steps that a shift-reduce parser goes through when it parses (x, x, x) . That is, show the contents of the stack, and the remaining input at each step.
4. Suppose that we replace the left-recursive production $L ::= L, S$ with a right-recursive one $L ::= S, L$. What general effect does this have on the depth of the stack during a shift-reduce parse? (*Hint: you might want to work through the parse of (x, x, x) again to see what changes.*)

Problem 3 (Cooper and Torczon, Problem 3.10)

Consider the following grammar:

$\text{Start} ::= S$

$S ::= A a$

$A ::= B C \mid B C f$

$B ::= b$

$C ::= c$

Is the given grammar an LR(1) grammar? Please show your work.

Problem 4 (Cooper and Torczon, Problem 3.11)

Consider a robotic arm that accepts two commands: ∇ puts an apple in the bag, and \triangle takes an apple out of the bag. Assume that the robotic arm starts with an empty bag.

A valid sequence for the robotic arm should have no prefix that contains more \triangle commands than it contains ∇ commands. For example, $\nabla\nabla\triangle\triangle$ and $\nabla\triangle\nabla$ are valid commands, but $\triangle\triangle\nabla\nabla$ and $\triangle\nabla\triangle$ are not.

1. Write a grammar that represents all the valid command sequences for the robotic arm.
2. Prove that the grammar is LR(1).